

In re: Fluke et al.
Serial No.: 09/607,074
Filed: June 29, 2000
Page 16 of 21

REMARKS

Applicants appreciate the Examiner's thorough examination of the present application as evidenced by the final Office Action of July 16, 2003 (hereinafter "Final Action"). Applicants, however, respectfully request that the Examiner take one final look at independent Claims 1, 9, 20, 28, 39, and 47. Applicants respectfully submit that the cited references do not disclose or suggest, among other things, invoking a print function with a format argument and saving the format argument in a deferred trace data buffer where the format argument is a pointer to a memory location in an address space of the application. Accordingly, Applicants submit that all pending claims are in condition for allowance. Favorable reconsideration of all pending claims is respectfully requested for at least the reasons discussed hereafter.

Information Disclosure Statement

The Office Action of January 21, 2003 indicated that the Brabec reference was not received with Applicants' Information Disclosure Statement filed June 29, 2000. In response, Applicants submit herewith a copy of the Brabec reference.

Independent Claims 1, 9, 20, 28, 39, and 47 are Patentable

Independent Claims 1, 9, 20, 28, 39, and 47 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U. S. Patent No. 5,983,366 to King (hereinafter "King") in view of the document "The Visual C++ Debugging Environment" authored by Keith Bugg (hereinafter "Bugg").

Independent Claims 1, 9, 20, 28, 39, and 47 are directed to methods, systems, and computer program products for printing data from an application in which a print function is invoked with a format argument and the format argument is saved in a deferred trace data buffer. The format argument is a pointer to a memory location in an address space of the application. Embodiments including this aspect of the present invention are described, for example, at page 11, lines 5 - 16 of the Specification.

The Final Action acknowledges that King does not disclose "the format argument being a pointer to a memory location in an address space of the application." (Final Action,

page 5). The Office Action does assert, however, that Bugg teaches "a format argument a debugging information output command being a pointer to a memory location in an address space of an application..." (Office Action, page 5). Furthermore, in rejecting Claim 10, the Final Action again acknowledges that King does not disclose "the format argument being a pointer to a memory location in an address space of the application, and saving the pointer in the deferred trace data buffer." (Final Action, page 11). The Final Action does assert, however, that U. S. Patent No. 6,282,701 to Wygodny et al. (hereinafter "Wygodny") teaches "displaying a pointer (for example, variable names) and the contents of the memory referred to by the pointer as part of a trace output display..." (Final Action, page 11).

To establish a *prima facie* case of obviousness, the prior art reference or references when combined must teach or suggest *all* the recitations of the claims, and there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. M.P.E.P. §2143. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. M.P.E.P. §2143.01, citing *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990). As recently emphasized by the Court of Appeals for the Federal Circuit, to support combining references, evidence of a suggestion, teaching, or motivation to combine must be **clear and particular**, and this requirement for clear and particular evidence is not met by broad and conclusory statements about the teachings of references. *In re Dembiczak*, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). In an even more recent decision, the Court of Appeals for the Federal Circuit has stated that, to support combining or modifying references, there must be **particular** evidence from the prior art as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed. *In re Kotzab*, 55 U.S.P.Q.2d 1313, 1317 (Fed. Cir. 2000).

Applicants respectfully submit that the King, Wygodny, and Bugg references contain no description therein to suggest to or motivate one skilled in the art to modify King's computer program tracing system with the teachings of either Wygodny or Bugg. In fact, Applicants respectfully submit that the disclosures of King, Wygodny, and Bugg teach

against such a combination as the resulting computer program tracing system as alleged by the Final Action would be inoperable.

King explains at column 19, lines 17 - 27, that a computer program may perform a trace by calling a trace macro, which in turns calls a trace function that includes a numerical identification of the trace message and a pair of parameters. In King's example, the trace message is identified as number 7292. The parameters and trace identification are packed into a message and transmitted from the data processing system 252 to the host processor 254. (King, col. 19, lines 40 - 44). Once the message is received at the host processor 254, the host processor 254 looks "up the trace id and then correctly unpack the trace message and display the trace string and its parameters in the way defined in the trace control file..." (King, col. 19, lines 50 - 52). Because King describes processing the trace message, which includes a trace identification and parameters, on a different processor (host processor 254) than the processor executing the computer program that is being traced (data processing system 252), replacing the trace identification with a pointer would not work because the host processor 254 does not have access to the address space of the data processing system 252. That is, a pointer to a memory location in the data processing system 252 is useless to the host processor 254.

In response to the foregoing analysis, the Final Action states that the "format argument pointers described in Bugg are inherently de-referenced prior to being output/stored..." (Final Action, page 3). The Final Action further states that "in the online debugger system of Wygodny, the host processor does have access to the client application address space..." (Final Action, page 3). For these reasons, the Final Action maintains that modifying the King reference with the teachings of Bugg or Wygodny would not render the King system inoperable. (Final Action, page 3). Applicants respectfully disagree with this interpretation of the King, Bugg, and Wygodny references.

As discussed above, because King describes processing the trace message on a different processor than the processor executing the computer program that is being traced and because the two processors do not have access to the same address space, King's computer program tracing system would not work if a pointer is used instead of the trace identification. This is because the pointer would be based on the address space of the data

processing system 252 rather than the host processor 254. Applicants further note that the claim language describes the format argument as being a pointer. Thus, it is irrelevant whether Bugg inherently teaches de-referencing a pointer first as the claim language does not recite the contents of a memory location pointed to by a pointer, but simply a pointer.

In response to the assertion in the Final Action that Wygodny discloses a system in which the host processor has access to the client application address space, Applicants respectfully disagree. As shown in FIG. 2 of Wygodny, the analyzer 106 and the client 102 share a trace buffer 105, but the analyzer does not have access to the address space of the user application executing on the client 102.

Accordingly, Applicants respectfully submit that one skilled in the art would not be motivated to replace the trace identification described in King with a pointer as described in Wygodny or Bugg as such a replacement would render King's computer program tracing system inoperable.

For at least the foregoing reasons, Applicants respectfully submit that independent Claims 1, 9, 20, 28, 39, and 47 are patentable over the cited references and that dependent Claims 2 - 8, 10 - 19, 21 - 27, 29 - 38, 40 - 46, and 48 - 57 are patentable at least by virtue of their depending from an allowable claim.

Dependent Claims 7, 26, and 45 are Separately Patentable

With regard to Claims 7, 26, and 45, these claims include all of the recitations from independent Claims 1, 20, and 39, respectively, and are, therefore, patentable over the cited references for at least the reasons stated above. In addition, Applicants submit that these claims are separately patentable as none of the cited references described or suggest "saving...a memory contents comprising the address space of the application in a non-volatile medium."

The Office Action alleges that Bugg teaches "sending debugging output, including format and data arguments to a file." (Final Action, page 3). Applicants respectfully submit that Bugg describes the ability to send a processed debug report, including a file name, linenummer, and a formatted message, to a file, debugger, or message window. The Final Action asserts that the filename, linenummer, and module name are indicators of the address

In re: Fluke et al.
Serial No.: 09/607,074
Filed: June 29, 2000
Page 20 of 21

space of the application. (Final Action, page 3). Applicants submit that whether the filename, linenummer, and module name may provide an indication of the address space or not is irrelevant. The filename, linenummer, and module name are not the actual address space of the application, which is a defined portion of memory where the application executes. Applicants submit that Bugg contains no description of saving the address space of the application in which the `_CrtDbgReport()` function is called to a non-volatile medium.

Accordingly, Applicants respectfully submit that Claims 7, 26, and 45 are separately patentable for at least these additional reasons.

In re: Fluke et al.
Serial No.: 09/607,074
Filed: June 29, 2000
Page 21 of 21

CONCLUSION

In light of the above amendments and remarks, Applicants respectfully submit that the above-entitled application is now in condition for allowance. Favorable reconsideration of this application is respectfully requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (919) 854-1400.

It is not believed that an extension of time and/or additional fee(s)-including fees for net addition of claims-are required, beyond those that may otherwise be provided for in documents accompanying this paper. In the event, however, that an extension of time is necessary to allow consideration of this paper, such an extension is hereby petitioned under 37 C.F.R. §1.136(a). Any additional fees believed to be due in connection with this paper may be charged to Deposit Account No. 50-0563.

Respectfully submitted,



D. Scott Moore
Registration No. 42,011

Customer No. 20792
Myers Bigel Sibley & Sajovec
P. O. Box 37428
Raleigh, North Carolina 27627
Telephone: (919) 854-1400
Facsimile: (919) 854-1401

Certificate of Mailing under 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop AF, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450 on September 15, 2003.


Traci A. Brown

RA888-0064

GRW

N. Brabec, J. L. Brodd, M. R. Cleveland and J. H. Maher



COPY

RECEIVED

SEP 24 2003

Technology Center 2100

32452

Triggering Scheme to Trap Trace Data

Many operating systems include facilities for tracing data which can be started and stopped under manual control. A better technique is to embed the capability to automatically halt the trace when specified conditions occur, assuring that the right trace information will be captured.

When problems occur in the microcode used to operate a hardware device, such as a communications controller, the end results frequently indicate that a specific failure has occurred. However, it is difficult to trace the events leading up to this failure. Manual attempts to stop a program trace at precisely the correct time often result in overflowed buffers, or truncated buffers with only partial information.

For example, it may be known that a task becomes suspended at a certain point. It may be known what the task's control block looks like when it becomes suspended. The desired trace would contain task activity up to and including the time when the task's control block indicates that it has been suspended. It is impossible to reliably use a manual method to stop the trace when the condition occurs.

A breakpoint facility has been designed which stops all tracing when a particular string of data occurs at a specific offset within the data block being traced. The facility also permits the criterion that the data string will appear in a specific kind of trace data. This function may be optionally selected by supplying the information in the form of parameters before the tracing begins.

For example, a trace may include tasks, communications data, and other data blocks. Indicating that the string of data will appear at a specific offset but must be within the task information block allows the trace to include communications information as well without fear of prematurely halting the trace.

When the string of data is found within the correct data block type, tracing stops. No further information is placed in the trace buffer. At this point, a complete trace has been captured as is available for analysis. The debug process is aided because the required information is automatically captured on the first attempt.

Disclosed anonymously
32452

Reproduced from
RESEARCH DISCLOSURE, April 1991, Number 324
© Kenneth Mason Publications Ltd, England